LucidDream: Dynamic Story Generation through Directed Chatbot Interactions

written by

Ryan Stonebraker

in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science

University of Alaska Fairbanks.

May 2020

Committee Members:
Dr Jonathan Metzgar, Committee Chair
Dr Orion Lawlor, Committee Member
Dr Chris Hartman, Committee Member/Department Chair

Abstract

Natural Language Understanding and Generation are both areas of active research with widespread potential for story telling. This paper proposes an architecture for dynamically generating stories that allows a scene to be constructed and then dynamically written through the interaction of individual chatbots. Each chatbot in this environment is meant to mimic either the specific emotional profile of a character or holistically represent all of the character's attributes. Chatbots are created using the conversation history so that they can understand context, a relevant sentence suggestion provided by a question-answering model to keep generated output on topic, and a fine-tuned version of the GPT-2 transformer-based language model to combine all of this information and generate text. This architecture serves as an ensemble method of approaching character modeling and also introduces the little-explored concept of emotional style transferring as a method for merging a story character's emotional attributes with an independent training corpus. The question-answering model used in this study achieved 65.24% accuracy when tested on the Stanford Question-Answering Dataset and the emotion classification model achieved 57.3% accuracy on the International Survey on Emotion Antecedents and Reactions dataset. While neither of these performances are SOTA for their respective individual tasks, they are used in combination to produce state of the art directed story generation and pave the way for future research.

Contents

Chapter 1

Introduction

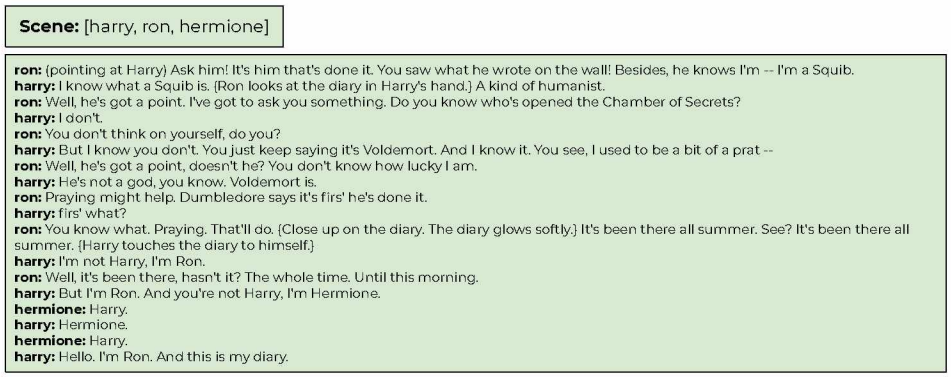

Figure 1.1: An example interaction between chatbots based on the Harry Potter characters Harry, Ron, and Hermione.

Communication is the first fundamental layer of intelligence from which all levels of higher thought are derived. As humans, it is the first thing we learn to do and it is one of the few thing that constantly evolves throughout our lifetimes. Verbal communication is the most primitive form of human communication and is coupled with intonation and body posture to reinforce

the intent that it carries. When thoughts are translated to written language however, all additional forms of intent indicators are stripped away and we are left with only one channel to carry meaning. Due to this, written communication is often much more verbose when conveying the same nuances, but is still full of ambiguity and context-derived meanings that occasionally confuse even human readers. Despite these challenges, written communication is one of the primary targets in recent years that we have attempted to get Artificial Intelligence systems to emulate. It is thought that if we are able to communicate with our computer systems in the same way that we are able converse with each other, then we will unlock a new era of possibilities where computers can become a much more natural extension to humanity.

Despite the extensive potential of generalized natural language chatbots however, chatbots for the most are often created with domain-specific purposes in mind. They are used to automatically provide customer support for companies, to return information from internet databases, and to provide canned, comedic responses to specific prompts. While this command-driven approach to building chatbots is meant to provide them with purpose, it in practice drastically limits their ability to naturally interact with users and leaves users frustrated when a non-scripted request arises.

The reason for the limited forms of chatbot models is not for lack of people trying to develop more sophisticated systems. In many ways it is known that a fully generalized, artificially intelligent chatbot would be the holy grail of not just Natural Language Understanding (NLU), but of all AI research.

Such a chatbot would provide a unified system for companies to use to provide support to customers and handle routine tasks that involve human interaction. Therefore, the lack of generalized models is instead systemic of the difficulty that this problem poses. A fully generalized model would have to not only understand a prompt, but also contextualize the prompt with past conversation, understand how this response should emotionally predispose the agent, and then actually perform some sort of Natural Language Generation (NLG) to generate an appropriate response. Achieving what would be considered good results by a human standard at any one of these tasks would be ground breaking research in itself, so instead, this paper aims to create a pipeline for combining existing State of the Art (SOTA), or near SOTA, methods so that current progress in this area can be evaluated.

Before diving into the implementation of the aforementioned pipeline, it is important to clarify both the question this paper aims to answer and the scope of research that this entails. Broadly speaking, the purpose of this paper is to determine whether modern advances in deep learning can be used to meaningfully combine question-answering and directed personality language-modeling in an unstructured environment. In order to quantify what meaningful means in this context, metrics evaluating the accuracy of question-answering, emotional profiling, and generated emotional bias will be used. Each of these metrics will be explained more in depth later in the paper, but it is important to clarify beforehand the goals which this paper aims to achieve. All associated code for this paper backing up these metrics is publicly available on GitHub [1].

Chapter 2

Prior Work

Natural Language Understanding, and subsequently its inverse twin Natural Language Generation, are relatively new fields that aim to apply advances in deep learning and neural networks to the medium of text. Some of the most groundbreaking papers in these fields, such as Google's paper on Bidirectional Transformers for creating better word embeddings [7], came out in only the last few years. Due to this, we are only now seeing the real-world applications of these techniques. With this in mind, it is important to not only evaluate specific related uses of NLU, but to also look at the parent field of Natural Language Processing (NLP), which has historically used hard-coded priori knowledge about linguistics to achieve results that we are just now seeing through deep learning alone.

```
1 Mary moved to the bathroom.
2 John went to the hallway.
3 Where is Mary? bathroom
4 Daniel went back to the hallway.
5 Sandra moved to the garden.
6 Where is Daniel? hallway
7 John moved to the office.
8 Sandra journeyed to the bathroom.
9 Where is Daniel? hallway
```

Figure 2.1: A sample from the Facebook bAbI Dialog-based Language Learning dataset illustrating the question-answering problem.

2.1 Question-Answering

Question-Answering is arguably one of the most important, yet complex problems that NLU aims to solve. To clarify, question-answering in this context refers to determining the answer to a question given a series of statements, of which one is assumed to host the answer, such as illustrated in Figure 2.1. In the past, this problem has been dealt with by using simple NLP techniques such as stripping stop words and matching regular expressions to the lemmatized series of words or Part of Speech (POS) tags. However, these simple approaches require language to be structured in an absolute way that cannot be guaranteed. Any deviation from the coded patterns leads to either false information being encoded or critical information being missed altogether.

Many tools exist for the previously mentioned type of deconstructive analysis, most notably the Python Natural Language Tool Kit (NLTK) library. However, it is not hard to find an ambiguous sentence that NLTK can't eas-

ily handle without hard-coding all permutations of text patterns. For this reason, libraries such as spaCy have recently been on the rise that aim to combine traditional NLP approaches with machine learning for tasks such as Named Entity Recognition (NER) and introduce the notion of processing pipelines to simplify tasks. When applied to question-answering, this provides further power to transform a natural language prompt into a thoroughly deconstructed and labeled input layer that can have the answer simply extracted based on its POS tag or position in the formatted layer. While this is great for domain-specific situations in which the types of questions and general structure of input can be guessed, it does not solve all generalized situations. When dealing with generalized situations in which not much can be assumed, some form of machine learning or NLU is required. However, this can be incredibly difficult in a completely unstructured environment where no metrics or feedback exist to even determine whether a response was correct or contains a reference to the correct answer. For this reason, researchers have largely turned to building labeled datasets to provide their models with at least some form of weak supervision.

2.1.1 Datasets

Two of the most notable datasets for question-answering are the Facebook bAbI (pronounced "baby") dataset and the Stanford Question Answering Dataset (SQuAD). These datasets are slightly different and are tailored towards specific approaches at solving this problem, but in general, they both aim at providing some form of supervision for models applied to this task. Both of these datasets have been widely used as testing grounds for SOTA

models and represent a great deal of human effort that was required to manually create and quality check the pairing of each answer to a question.

The Facebook bAbI Dialog-based Language Learning dataset [11] is only part of a larger effort by Facebook to encourage building models that learn in a similar way to human babies [9]. The dataset is structured as a series of simple stories in which statements continuously build off of each other and describe an evolving scene. In the midst of these statements, questions are interjected asking about the current state of events, as can be seen in Figure 2.1. The questions are then followed by an exact answer to each question. While this dataset can be used independent of the rest of Facebook's project, it was intended to only be one of many supervised environments for ML and AI agents to be taught how to answer contextual questions. Accompanying this dataset, Facebook proposed a series of other tasks and datasets these agents could be trained and tested on and released much of the code open source on GitHub. Another interesting proponent Facebook outlined in a related paper was a Human-in-the-loop (HITL) dialogue simulator [12]. This simulator provides a framework under which reinforcement learning agents can be expanded past the other fixed datasets and trained alongside the guidance of a human dialogue partner to be fine-tuned for their specific designated environments. While this paper will not cover a related approach using reinforcement learning, it is important to note the different techniques that have been proposed to solve the same task.

Since the Facebook bAbI tasks were outlined in 2015, there have been many

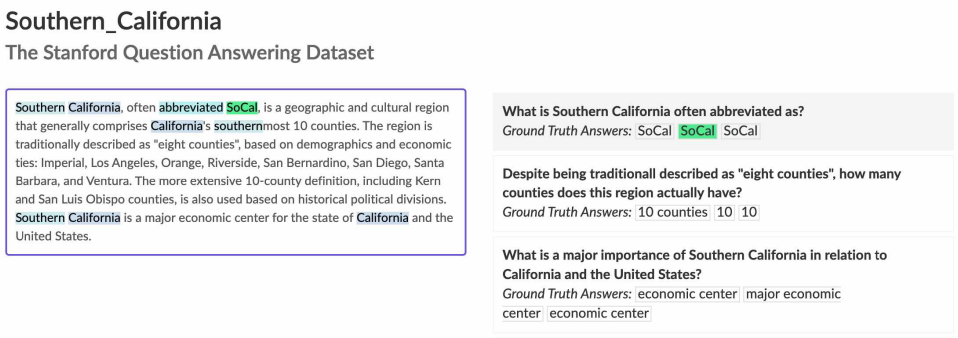

Figure 2.2: A sample from the SQuAD 2.0 dataset showing the type of questions in the dataset and highlighting the type of comprehension required.

attempts at achieving the highest possible accuracy for each one of them on the provided datasets. Due to the main dataset, the Dialog-based Language Learning dataset, being simplistic and limited in size because it had to be hand-curated by humans, many researchers have since achieved 100% accuracy on the tasks. In one paper published by the founder of the AI company Pat Inc, 9 of the 20 total tasks were attempted and 100% was achieved on each [10]. However, this paper goes on to point out some of the problems with the dataset such as answers that change based on context interpretation and ambiguous exact-answer phrasing. Such problems are systemic of all natural language and the fact that they crop up in an intentionally simplified subset of question-answering problems highlights the difficulties of the problem as a whole.

While Facebook aimed at providing a simplified, ambiguity-reduced environment in which question-answering models could be trained to perfect or near perfect accuracy, the Stanford Question Answering Dataset [13] takes

a different approach. Instead of toy contextual stories, SQuAD is composed of questions crafted from Wikipedia articles by crowd workers. By using Wikipedia articles, questions were able to be crafted that had less ambiguous answers, but required some reading comprehension to deduce, as shown in Figure 2.2. Human performance at this task in a study done by Stanford came out to an exact-answer match accuracy of 86.831%, which is better than the 51% accuracy of a logistic regression baseline approach outlined in the original paper [13], but suggests that the problem is sufficiently complex even for a human. Due to the format of the dataset, humans can essentially be treated as a competing agent solving the problem and are not necessarily the most efficient at the task. This is proven by the current top performer on the SQuAD dataset, an ensemble ALBERT + DAAF + Verifier approach with an exact match accuracy of 90.386%, which is notably higher than human performance.

2.1.2 Modeling

One of the simplest and oldest forms of question-answering involves building a bag of words (BoW) language model and performing cosine similarity. This approach simply encodes each word seen in a training dataset incrementally to build a language model (BoW) and then compares the numerically encoded question to each numerically encoded statement and chooses the most similar one based on the cosine similarity between the two matrices. Done naively as described, this approach suffers from not understanding context and over-weighting grammatically insignificant stopwords and other domain-specific words commonly found throughout the entire dataset.

Despite the poor performance of a naive implementation of BoW and cosine similarity for question-answering, there are many variations that dramatically improve results. One of the simplest additions to this technique is applying term frequency-inverse document frequency (TF-IDF) to the bag of words language model. This lowers the weight of words that are frequently seen throughout the dataset and thus makes less common and more descriptive words weighted as more important. Using this simple technique, over-weighting issues due to stopwords and repetitive words are significantly alleviated and performance is improved. However, this approach still leaves out a majority of localized contextual information.

The problem of building a performant question-answering model turns out to nearly entirely be a problem of building a more sophisticated language model that can adequately encode context. TF-IDF primitively encodes contextual information by devaluing non-important words, but isn't sufficient for encoding complex relationships. Building contextually aware language models has been one of the most cutting edge areas of research in NLP in recent years and new variations on models such as BERT [7] have been responsible for driving new high scores on datasets such as SQuAD. However, in order to properly build off of the success of these models, it should first be understood how they were derived.

One of the first major advances in language modeling was the creation of GloVe (Global Vectors) in 2014 [5]. This unsupervised language model-

ing framework creates word embeddings using techniques for global matrix factorization and constructing local context windows. Global matrix factorization involves reducing the complexity of large term frequency matrices and is done largely to improve compute time of building the embeddings on large training text. A local context window is used to represent the relationship of nearby words throughout the language corpus. GloVe relies on two methods for constructing a local context window; Continuous Bag of Words (CBOW) and skip-grams. These methods, while both aiming to incorporate context into the language model, do so in complete opposite ways. CBOW uses words surrounding a target word (the context window) to try and predict the target word and encodes the context words based on this information [14]. A skip-gram on the other hand tries to predict the context in which a word occurs based on the target word and encodes the target word based on this information. By using both of these strategies, GloVe is able to achieve excellent performance at establishing localized etymological relationships for all words it has seen.

Despite GloVe's great performance on recognizing relationships between most words, GloVe and related approaches that rely on whole-word vectorization fail at recognizing misspellings or words that weren't seen in training. In order to compensate for this lapse, approaches such as fastText exist [6]. FastText works by first breaking down whole-words into partial n-grams of finite character length. It then uses a skip-gram model to establish context between these pieces. This approach is much simpler than GloVe and does not work perfectly in every respect, but it performs great on words that

it has never seen before as it can infer their meaning based on the partial n-grams that compose them. For example, if the word "running" was in the language dataset, fastText could infer its relation to the word "run", whereas GloVe would treat it as an entirely different word.

The two outlined approaches above, GloVe and fastText, represent giant leaps above the methods that were considered SOTA only a decade ago. However, GloVe and fastText are by no means the end of the line. Both of these methods are flawed in some way and while they take great steps towards understanding context, they suffer from the problem of polysemy where the same word can portray different meanings in different contexts. To combat this problem, the idea of directional context was conceived where the same word is encoded differently based on the words proceeding it or following it. ELMo (Embeddings from Language Models) represents one of the first major frameworks that took advantage of this concept by using a bidirectional LSTM to encode context [15]. A successor to ELMo, ULMFiT (Universal Language Model Fine Tuning), took the ideas that ELMo had introduced and applied them at a much larger scale on the Wikitext-103 dataset to build a massive pre-trained model. This pre-trained model can then be fine-tuned towards domain-specific applications with only a few examples. This introduced the concept of transfer learning, which has had great success in image processing, to the medium of text [17].

In 2017, Google researchers released arguably the most pivotal paper in modern NLP and NLU [18] regarding the transformer architecture which uses

attention mechanisms to extract important contextual information. This had numerous wide-spread implications, but most importantly, it influenced the way language models were created and led to the rise of BERT (Bidirectional Encoder Representations from Transformers). BERT combined the previously mentioned concepts of directional context, the transformer architecture, and pre-training. Since its inception, BERT has been shown to be almost unreasonably effective at a variety of NLP tasks, including question-answering. On the SQuAD leaderboard, nearly all of the top approaches are variants of BERT. The most recent (non-ensemble) variant being ALBERT (A Lite BERT) [3], which is essentially a parameter reduced version of BERT that is more scalable and was able to be pre-trained on a 16 GB combined text corpus to achieve SOTA results.

2.2 Text Generation

Natural Language Generation (NLG) is often referred to as a the inverse of NLU. Instead of understanding text that exists, this subset of NLP is concerned with generating new text. However, in order to generate text intelligently, an understanding of context and intent needs to be known, making NLG actually very related to NLU and in practice generally utilizes the same techniques. Therefore, it follows that one of the most primitive forms of text generation is also one of the most primitive forms of NLU, BOW and cosine similarity. Instead of using this approach to encode language though (or more accurately, in conjunction with an existing language

model), NLG directly takes the previous sentence or question, encodes it, and then uses the most similar statement or word(s) as a response. As can likely be assumed though, this approach is not very effective at understanding previous context or intent and can only combine existing sentences or words.

Luckily for NLG, the advances that have propelled NLU forward also apply to text generation. Specifically, OpenAI's GPT (Generative Pre-Training) and GPT-2 architectures have been able to achieve stunningly human-like output by taking advances in language model and tuning them to work in a predictive fashion. GPT-2 in particular highlights the effectiveness of a transformer architecture when applied for generative purposes. Previous approaches at text generation utilized variations of different language modeling tactics such as ULM-FiT or GloVe paired with character-level RNNs to understand basic syntax and generate text. However, these approaches pale in comparison to a transformer based architecture that can not only demonstrate an understanding of context, but can use attention mechanisms to make output appear to have directed intent. While much credit is due to OpenAI for highlighting this with their GPT architectures, it is important to note that this architecture is essentially a summation of SOTA research on transformers and did not introduce anything fundamentally outside of this scope.

	idf_weights		idf_weights		idf_weights		idf_weights		idf_weights		idf_weights		idf_weights
anger		disgust		fear		guilt		joy		sadness		shame	
angry	2.764360	disgusted	2.982174	night	2.882068	felt	2.649230	friend	3.129147	died	2.751048	ashamed	2.762346
friend	2.928471	saw	3.059951	one	3.099133	guilty	2.737142	got	3.251749	friend	2.839996	felt	2.859288
one	3.340325	felt	3.290475	afraid	3.177914	friend	2.863747	time	3.371709	sad	2.869496	friend	3.189790
got	3.615268	people	3.299777	alone	3.245736	mother	3.266149	passed	3.381561	felt	3.167209	one	3.423404
time	3.640911	one	3.440007	fear	3.318496	one	3.424898	felt	3.432333	time	3.476631	time	3.558441
told	3.660651	friend	3.483970	home	3.318496	time	3.588050	first	3.442805	away	3.556673	told	3.673216
felt	3.736221	man	3.565887	car	3.347206	told	3.680831	happy	3.474893	father	3.580771	mother	3.850147
person	3.841562	disgust	3.655118	would	3.407224	went	3.767843	joy	3.530773	death	3.618042	people	3.850147
people	3.873842	person	3.878262	felt	3.449335	something	3.798514	university	3.665948	one	3.670002	found	3.866407
someone	3.890372	drunk	3.947255	time	3.563094	would	3.846624	school	3.692616	close	3.697031	something	3.866407
mother	3.907179	someone	3.983622	could	3.664447	home	3.863153	good	3.692616	mother	3.724810	class	3.882937
friends	3.907179	friends	4.002314	friend	3.762427	parents	3.879960	one	3.706222	years	3.753384	friends	3.882937
day	3.941665	time	4.080786	going	3.823052	day	3.897054	year	3.734002	friends	3.844356	day	3.899744
brother	3.959365	like	4.101405	house	3.823052	got	3.932146	friends	3.936238	home	3.844356	shame	3.899744
without	3.959365	girl	4.165944	back	3.838800	made	3.950164	exam	3.953630	left	3.909953	girl	3.988297

Figure 2.3: The top-words per emotion, as determined by TF-IDF, in the ISEAR dataset.

2.3 Evaluating Emotion

Not much work has been done on detecting general emotions in text. This is partially due to the difficulty in creating useful datasets for this task. One of the largest existing datasets, ISEAR (International Survey on Emotion Antecedents and Reactions), was created using crowd sourcing to label 2500 sentences with 7 different emotional categories [19]. While 2500 sentences is a good starting point, it is by no means a big dataset and in order for a supervised approach to adequately perform well using this data, pre-trained model transfer learning is a must. One of the reasons for the small size of the dataset is due to the ambiguity regarding emotions. Emotions are very nuanced and subjective are commonly misunderstood even among people. Due to this, each sentence in an emotional dataset like ISEAR has to be cross-validated multiple times in order to have any semblance of accuracy by popular opinion. To reduce cultural bias, ISEAR was created on an international scale to average out emotional responses. However, while averaging

out emotional responses may help in creating a less-subjective quantification of emotion, it does not account for emotional locality in which things become a certain emotion only because of their preceding context or specific cultural origins. To properly gauge such emotion would require a deep understanding of the language style and good contextual awareness. Even if the mentioned conditions could be satisfied and a model constructed with a high level of contextual awareness and understanding, it would be likely that it would discover more than just seven emotional groupings. These however are the limitations provided by ISEAR and for the purpose of this paper, we will assume that ISEAR can provide us with a good enough emotional estimator to use as a starting point.

Chapter 3

Implementation

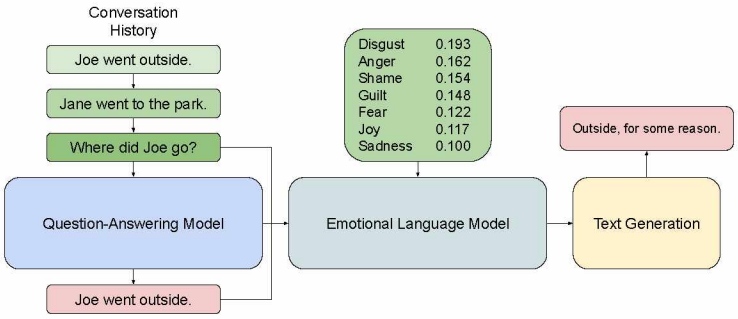

Figure 3.1: A high-level overview of the response pipeline that uses the sentence containing the answer to a contextual question as a seed (along with past conversation) and passes this to an emotional language model that is used to generate text output emulating a character profile.

The goal of this paper is threefold. First, a question-answering model will be created using near SOTA methods. Next, fine-tuned language models will be created that will take the question-answering model output as a suggestion along with an entire history of conversation to guide the text generated and ensure it stays contextually relevant. Lastly, the outputted text will be

sent to an environment in which chatbots with different emotional profiles and trained on different texts will accept the output as input and respond accordingly following the pipeline seen in Figure 3.1. In addition to chatbot-chatbot interaction, this environment will also be constructed to allow for human-chatbot interaction.

In order to create a question-answering model, the SQuAD dataset will be used. The top performing approach at question-answering on SQuAD is currently an ensemble architecture that uses ALBERT to understand context and extract specific answers. While this is great for question-answering in the traditional sense, the purpose of the question-answering model for this paper is to seed the emotional language models. Using only a single-word seed presents the risk of ambiguity as a language model could find many different directional contextual instances of a single word. Therefore, the entire sentence containing the answer will be used as a seed. In order to determine the sentence containing the answer to a contextual question, a fastText language model trained on SQuAD will be used. The reason fastText was chosen is due to its simplicity and its ability to perform quickly in real-time. Additionally, fastText is able to assume some understanding of words that were not found in its base training dataset, which is extremely useful in more generalized contexts. These performant qualities of fastText will then be combined with Facebook's InferSent architecture [2] to create sentence embeddings that cosine similarity can be used on. An example of this is shown in the simple scenario in Figure 3.2. As can be seen in this figure, the word "walked" does not appear in the visualizations of word im-

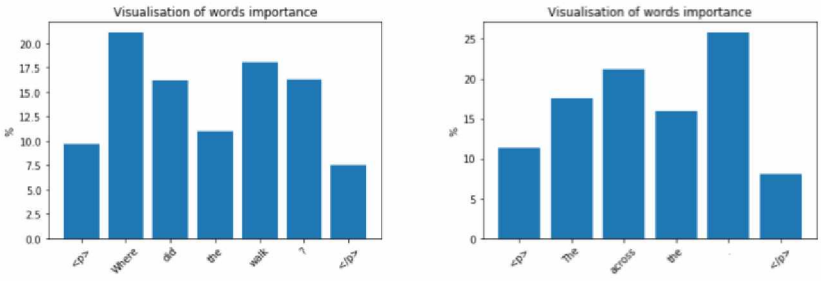

Figure 3.2: A simple question-answering model using fastText and cosine similarity that was trained on the SQuAD dataset and shows the perceived importance of each word in the question and highest scoring answer.

portance. This is presumably because the word "walked" does not appear in the SQuAD dataset where this model was trained. Despite this, fastText still chooses the correct statement as the answer because the word "walk" shares enough common partial n-grams that it is able to assume a relationship. The results of fastText on SQuAD will be discussed later in the paper.

The next major part of this paper is the construction of emotional language models that can produce output emulating a character profile. Characters from a Harry Potter will be used as target profiles and movie script dialogue from each character will be analyzed using a model trained on the ISEAR dataset to determine a categorical emotional breakdown. These profiles will then serve as basis of comparison for emulated model output. For the composite model that will be tested, these profiles will additionally be used to construct a corpus based off of the ISEAR dataset that matches the

emotional breakdown percentages. The emotional language models for all approaches will be fine-tuned instances of the SOTA GPT-2 model. Through this multi-step process, an emotional style transfer will be performed that creates a chatbot emulating a Harry Potter character's emotions that can be used later in the chatbot environment. The error margin for generating responses matching a criteria is expected to be quite large as it will be entirely reliant on the ISEAR model being able to properly identify emotions.

Figure 3.3: A diagram highlighting how character presence in an unlabeled dialogue corpus is determined. In this example, n=3 for determining which characters are present in the scene, where n is the number of subsequent conversational exchanges in which the character does not speak.

The last step needed after chatbots are created is an environment in which they can interact. This environment will be constructed so that both chatbot-chatbot and human-chatbot interaction can occur. Under this scheme, a human is essentially treated as another chatbot that helps guide conversation. The history of all conversation in the environment will be used as the primary source for the question-answering model. After picking a sentence most likely to contain an answer to a contextual question, the question-

answering model will have its output provided as a suggestion for the text generation model to use. In order to prevent responses being constrained to the same pool of shared statements, an individual character model will be created that will have its own respective history of dialogue for each character. For the Harry Potter models, this will come from the dialogue in which the character was present in the scene. Presence in a scene will be determined based on a fixed proximity to a response by the character as shown in Figure 3.3. While this presence technique is not completely accurate, it should be good enough to generate a sufficiently large, unique, and ordered corpus for each character. Lastly, in order to direct the sequence of interactions, an environmental agent will be added as well. The environmental agent will essentially be equivalent to a regular chatbot, but it will be holistically trained on all environmental statements in the first three Harry Potter movie scripts combined and should produce less opinionated and more factual based output.

Chapter 4

Results

In order to gauge the performance of the directed chatbot interaction, each component of this paper will be evaluated using discrete metrics. The question-answering model will be tested on the SQuAD dataset. Instead of using the standard exact-match metric used for models competing on the SQuAD scoreboard however, it will be evaluated on whether the exact-match is contained in the chosen sentence. The emotional language models will be evaluated on three levels. First, the emotional classification model will be directly compared to the crowd-sourced labels of the ISEAR dataset for accuracy. The composite emotional GPT-2 model will then have its emotional output evaluated by the ISEAR classifier. This metric will have the largest expected compounded error. Lastly, the output of the individual and holistic emotional models will have their generated emotional profiles created using the ISEAR model and this will be compared against each respective character's emotional profile from the movie scripts. This will be used to determine the accuracy of the emotional style transfer.

4.1 Question-Answering Model

FastText is a relatively simple model when compared to many of the approaches used on the SQuAD dataset. Despite the simplicity however, it was able to correctly identify the sentence containing the answer to a given contextual question 65.24% of the time. This metric was computed by iterating over each question in the SQuAD training dataset and running the model against the associated statements. It is important to note that this accuracy is not for exact-answer matching and is on the training SQuAD dataset (versus the larger dataset) and therefore is not necessarily comparable to those found on the SQuAD explorer. Nevertheless, 65.24% is sufficient for the application of seeding an emotional chatbot's response. It will be left for future work to experiment with how a more sophisticated model influences generated output.

In regards to the text generation pipeline, the question-answering model was used in conjunction with prior conversation to seed the emotionally biased text generation model. The thought behind this was that it would influence generated output and provide a contextual grounding that would help keep the dialogue on track. However, it was found in many cases that the text generation model was able to adequately answer questions without this influence simply based on the inclusion of past conversation. It was difficult to evaluate how many times this was the case however as the wording varied and it would be non-trivial to automatically identify whether a response adequately answered a question.

4.2 Emotion Modeling

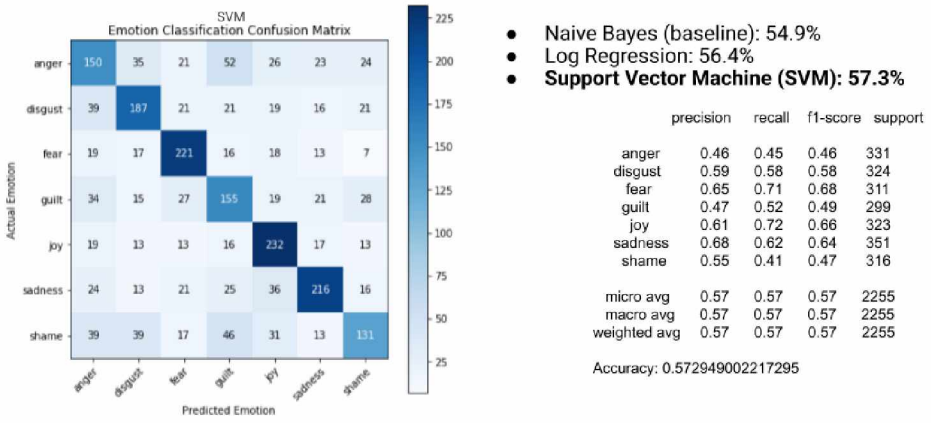

Figure 4.1: The confusion matrix for a simple SVM model applied to the ISEAR dataset.

In order to classify emotional sentiment, the statistical Naive Bayes, Log Regression, and Support Vector Machine (SVM) algorithms were used as shown in Figure 4.1. Naive Bayes is a common baseline approach that assumes each feature used is independent and generatively models the joint distribution of a feature compared to the label. While this model is great on small datasets and situations in which feature independence can be verified, it quickly deteriorates as the size of the feature space increases, such as is the case when comparing large word embeddings. Nevertheless, it was able to achieve an accuracy of 54.9% at classifying emotional sentiment on the validation set of the ISEAR dataset.

The next common text classification model is logistic regression. Logistic

regression often performs better than Naive Bayes at text classification by using a discriminative model to approximate the class decision boundaries. Another benefit of logistic regression is that it provides a percent confidence breakdown for each class, which is extremely useful when trying to understand a model. The trade off when compared withe Naive Bayes is that it takes slightly longer, but when dealing with a small dataset the size of ISEAR, this is negligible. This model was able to achieve 56.4% at identifying emotional sentiment, which while still not great, is an incremental improvement.

The last model evaluated for determining emotional sentiment in ISEAR was an SVM. An SVM works by attempting to find optimal decision boundaries that maximizes the distance between any given points, or word embeddings in this case. Again, the trade off with using an SVM over Naive Bayes is that it is slightly slower, but this is not a problem for this case and took only a couple minutes to train. The model was able to achieve 57.3% accuracy at classifying emotion. While this is the highest out of all three methods, it is close enough to logistic regression where this model was chosen for the benefit of its emotional confidence breakdown.

4.3 Character Profiling

After baseline performance was established for the question-answering and emotion classification models, the next step was to begin building character based language models. For this purpose, seven essential Harry Potter

harry	percent	hermione	percent	ron	percent	albus dumbledore	percent	snape	percent	hagrid	percent	tom riddle	percent
disgust	0.164353	disgust	0.160070	disgust	0.165744	fear	0.172254	disgust	0.161032	shame	0.159073	fear	0.221062
anger	0.158511	fear	0.159020	anger	0.158743	anger	0.167840	anger	0.159126	disgust	0.156533	anger	0.167995
fear	0.154489	shame	0.153700	fear	0.153267	disgust	0.155792	guilt	0.157026	anger	0.151317	disgust	0.137993
guilt	0.148062	anger	0.152627	shame	0.147874	shame	0.147213	shame	0.151813	guilt	0.146001	shame	0.133427
shame	0.145720	joy	0.139356	guilt	0.147454	guilt	0.145414	fear	0.150802	joy	0.142905	guilt	0.126713
joy	0.122984	guilt	0.138728	joy	0.124697	joy	0.115502	joy	0.122315	fear	0.142477	joy	0.115119
sadness	0.105881	sadness	0.096499	sadness	0.102221	sadness	0.095985	sadness	0.097886	sadness	0.101693	sadness	0.097691

Figure 4.2: An emotional breakdown of each Harry Potter character as determined by the ISEAR classification model.

characters were analyzed to determine their emotional profile, shown in Figure 4.2. In order to create each profile, every line in the first three Harry Potter books that was associated with a specific character was run through the logistic regression ISEAR classifier and the mean score for each emotion was outputted. It is important to note that when looking at these numbers, emotion is being accurately classified approximately 56.4% of the time (based on the ISEAR validation data, which is not guaranteed to be generalizable). Adding to this error, the word embeddings are being computed simply using TF-IDF, which does not provide extensive contextual information. Therefore, in a book setting such as Harry Potter where emotion is often derived by events that are contextually taking place, accuracy can be expected to be lower. Nevertheless, these profiles give us some semblance of ground truth data in terms of what we should be looking for from the generated chatbot output.

After emotional profiles for each character were constructed, three text generation approaches were explored using a fine-tuned version of the GPT-2 algorithm. The first approach involved further training a GPT-2 model on text specifically from each category of emotion in the ISEAR dataset.

harry	percent	ron	percent	hermione	percent	snape	percent	albus dumbledore	percent	tom riddle	percent	hagrid	percent	harry_as_voldemort	percent
joy	0.185362	disgust	0.182333	guilt	0.202032	shame	0.240175	shame	0.169533	shame	0.179887	fear	0.202330	anger	0.167776
fear	0.167128	anger	0.161311	joy	0.160415	fear	0.181152	anger	0.160235	fear	0.176536	shame	0.166199	fear	0.166763
shame	0.157771	guilt	0.157838	anger	0.152382	anger	0.144979	fear	0.157863	anger	0.166075	anger	0.149175	disgust	0.164662
guilt	0.134952	joy	0.146487	disgust	0.137001	disgust	0.124236	joy	0.138714	joy	0.150050	joy	0.132464	shame	0.148685
disgust	0.129723	shame	0.131529	shame	0.136766	guilt	0.121135	disgust	0.135560	guilt	0.133627	disgust	0.129813	guilt	0.133939
anger	0.125723	fear	0.118992	sadness	0.123010	sadness	0.099022	guilt	0.134512	disgust	0.113813	guilt	0.123038	joy	0.124216
sadness	0.099341	sadness	0.101512	fear	0.088394	joy	0.089301	sadness	0.103582	sadness	0.080012	sadness	0.096980	sadness	0.093960

Figure 4.3: An emotional breakdown of each holistic model generated Harry Potter character chatbot using the ISEAR classification model.

These single-emotion models were then used in a composite fashion according to the profile outlined for a target character in Figure 4.2. Output was constrained to a maximum of 30 words and split up into random length segments, with each segment having the probably of being a certain emotion dictated by the profile. In theory, averaged out over time and assuming each single-emotion model only outputted text that identified as it's associated emotion, this would yield generated text exactly matching the character's emotional breakdown. As shown in Figure 4.3 and additionally in Figure 4.4 however, the outputted profiles for each character, while similar, are not an exact match.

	character	book_similarity		character	book_similarity		character	book_similarity
composite			individual			holistic		
0	harry	0.906905	0	harry	0.978971	0	harry	0.976888
1	ron	0.724984	1	ron	0.978439	1	ron	0.990880
2	hermione	0.908331	2	hermione	0.993640	2	hermione	0.960975
3	snape	0.736106	3	snape	0.955563	3	snape	0.962648
4	albus dumbledore	0.924878	4	albus dumbledore	0.970592	4	albus dumbledore	0.993733
5	hagrid	0.942674	5	hagrid	0.981586	5	hagrid	0.982585
6	harry_as_voldemort	0.683473	6	harry_as_voldemort	0.729079	6	harry_as_voldemort	0.986740
7	tom riddle	0.892414	7	tom riddle	0.868134	7	tom riddle	0.981684

Figure 4.4: Each character's generated output cosine similarity compared to their calculated book profile. The cosine similarity was computed using TF-IDF vectorization and the harry_as_voldemort character is compared to the target Tom Riddle emotional profile.

The next approach used for text generation was that of an individual character model approach. Under this approach, the three-book Harry Potter corpus was analyzed and had an additional field of character presence added as previously defined in Figure 3.3. The corpus was then split up for each character based on their presence in the scene and this new generated corpus was used to fine-tune respective GPT-2 models. The outputted text of this model aligned with the perception of the respective Harry Potter character much better than the composite approach due to specifically being trained on their language style, but as can be seen in Figure 4.4, it was actually not better at outputting the same emotional profile for all characters such as Tom Riddle.

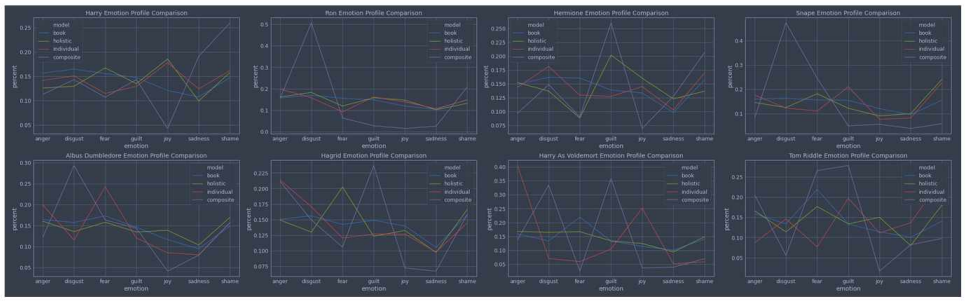

Figure 4.5: Emotional profile comparisons for each character and modeling approach.

The last approach used for text generation was a holistic one. This approach involved training a GPT-2 model on the entire three-book Harry Potter corpus and then calling it to produce output for each character when the character was selected to respond. It is important to note that while this approach has holistic information about all characters, when it is called,

it is coerced into generating text specifically for an individual character. As shown in Figure 4.4, this performed well for all characters.

In order to better understand the emotional profiles of the text generated by each approach, Figure 4.5 aims to compare the shapes of each graph against the ground truth characters. As can be seen, there is a lot of variance across the models despite the seemingly high cosine similarity. It should also be noted that the harry_as_voldemort character was created by training an individual character model on a subset of text from the Harry Potter character that was constructed following the emotional profile of Tom Riddle. For example, Tom Riddle was shown to have text associated with the shame emotion 22% of the time, so 22% of the text used to build the subset corpus was also identified as the shame emotion by the ISEAR model, but spoken by Harry. Due to the nature of this emotional style transfer, only the composite and individual character models would be expected to perform adequately well. However, Figure 4.4 actually shows the opposite and that the holistic model far outperformed the individual and composite models at matching the emotional profile. This could possibly be because the GPT-2 model is provided with the character name (harry_as_voldemort), and since it contains the names "harry" and "voldemort", is able to produce text that is most commonly associated in context with both of these characters, thus accidentally performing an emotional style transfer simply by virtue of combining their names.

4.4 Environment Interaction

Figure 4.6: Sample output from the composite model approach. It is important to note that this model does not use the Harry Potter corpus and instead is just a representation of each character's emotional profiles.

One of the fundamental traits of a dynamic story generation environment is the interaction of multiple different chatbot agents. Many chatbot environments focus solely on responding to a given prompt and shirk off the need for any level of conversational awareness. Conversely, pure story text generation is often done by using a single model to maximize the level of coherence and awareness. The environment used in this paper lies in between these two extremes and supports the interaction of multiple agents, including a human user, but also emphasizes the use of a conversation history to guide responses. The agents that are present in a given scene can furthermore be chosen, which cannot be done if a single model is generating all of the output. All conversation is also started with a manually entered scene to direct the subsequent interactions.

Although none of the three text generation approaches perfectly capture the emotional profile of a given Harry Potter character, all are able to produce output that is contextually aware to a surprising degree. The best way to prove this though is through example. Figure 4.6 shows the sample output

generated by the composite models following the emotional profiles of all of the Harry Potter characters. It is important to note with this approach versus the two others that the training text is the ISEAR dataset. Therefore, it is not expected for this model to produce responses that sound like each individual Harry Potter character, but rather that have responses that match the same emotions as the respective Harry Potter character, almost as if this character were placed into another environment without any memory.

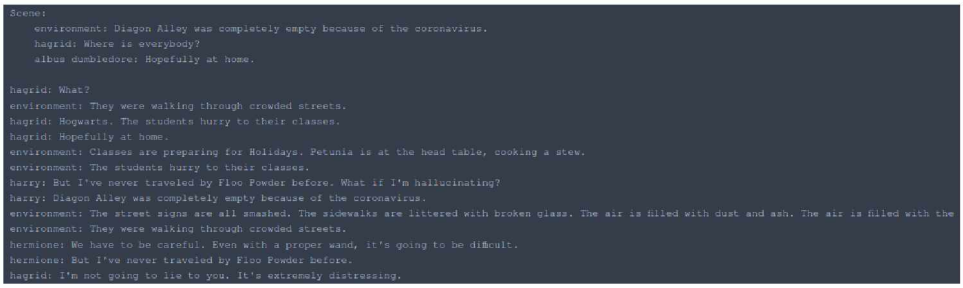

Figure 4.7: Sample output from a conversation generated with individual character models.

The individual character modeling approach did utilize the Harry Potter corpus and therefore all respective output has the expectation of belonging to the source text. A sample conversation using this model is shown in Figure 4.7. While output can be seen in this figure from the environment chatbot, the generator behind this was actually a holistic model, as it did not make sense to emotional classify text describing the environment. Interestingly though, it worked sufficiently well to simply treat the environment as another chatbot in order to gain the same level of contextual awareness.

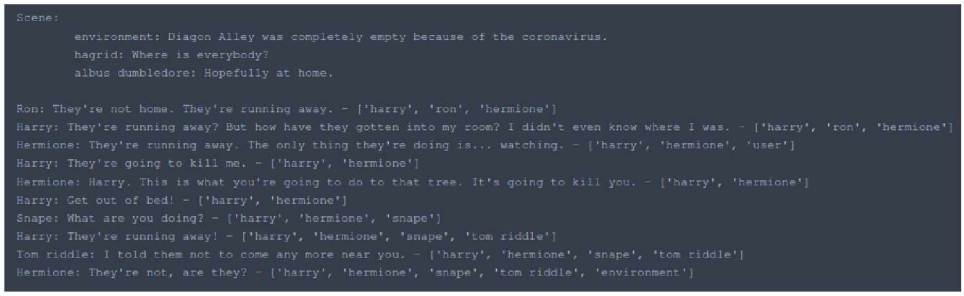

Figure 4.8: Sample output generated by a holistic model instance running for each character in the scene.

The holistic model proved to be arguably the most contextually aware, as is highlighted by sample output in Figure 4.8. The indented scene text was manually added as a base seed for the models to expand off of and includes a question and a speculative answer by a character. The first holistic model, in response to this scene text, is shown directly responding to this information in a contextually aware manner and changing the direction of conversation. One reason which could explain the increased contextual awareness as compared to the individual character models would be that since the corpus it used for training was not split up, it had access to more sensible sequential conversation than the individual models did. In order to make comparable individual character models, a more contextually coherent corpus focusing on each character would be needed. One of the main reasons it was initially speculated that an individual character model would provide better output was because a character could therefore not speak on information that they should have been unaware of (ie. because it occurred when they were not present in the text). However, the holistic model appears to perform well at separating this information out based simply on the character it is directed

to respond as.

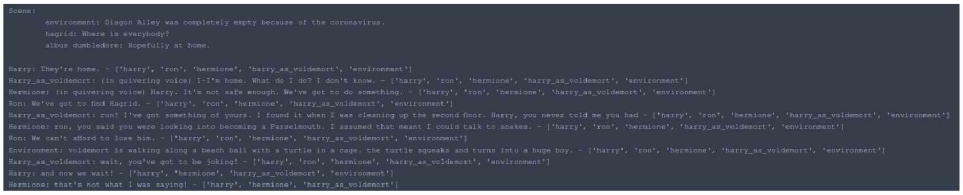

Figure 4.9: Sample output from individual holistic models and the emotional transfer model from voldemort to harry potter.

One of the most interesting areas of comparison in this study involves the output of the emotional style transfer characters. Figure 4.9 shows an example of output containing the harry_as_voldemort character, which utilizes an individual character model. The goal of an emotional style transfer is for the generated character to sound like the source character, but with the emotional characterization of the target character. In the example shown in Figure 4.9, the fear emotion seems to be the most prominent (as it is in Tom Riddle/Voldemort's profile). While this is arguably primarily a qualitative assessment when looking at specific examples and open to much interpretation, it opens the door to a not widely explored area of research.

Chapter 5

Discussion

The primary goal of this study was to determine whether SOTA deep learning could be leveraged to combine question-answering and emotional language modeling in a meaningful way to dynamically generate stories. While there is much ambiguity surrounding the interpretation of "meaningful" even when specific, quantitative data is provided supporting the accuracy of each model, there is much support for this hypothesis.

One of the largest problems faced in this study was collecting a sufficiently large and unambiguous corpus of text with labeled emotional sentiment. ISEAR was primarily used for this purpose, but was far from perfect. A degree of interpretation existed for almost all of the labeled text from ISEAR and given additional contextual information, the interpreted emotional connotation could easily change. While this could be argued to be a fundamental problem with the very idea of labeling emotions, the dataset is lacking in many other regards as well. For one, the text is largely written in first

person and contains personal attribution of feelings. This limits the scope of how it can be used as it does not contain much or any third person or situation-based emotion examples. Microsoft mentions this problem in an article from 2015 [4], but in the time since then, it does not appear that much work has been done to solve it. In practice, this could be seen as indicative that emotion is a byproduct of contextual awareness and not something that can be simply labeled with a single emotion. If this is the case, it could be beneficial to use a sophisticated language model such as ALBERT to try and understand emotion from context, rather than create a text generation model with specific emotional knowledge already in mind. Additionally, the emotion categories outlined by ISEAR are rather limiting and the inclusion of more could actually prove to be beneficial for more accurate classification.

Another problem that came up during this study was related to the inconsistencies in the text that was used to fine tune the GPT-2 model. In order to create an accurate model, a corpus of text was needed that was attributed not only to an individual character, but also to the environment. Movie scripts contain this type of attribution and were therefore what they Harry Potter dataset was created based on. However, there does not appear to be any consistent data format for movie scripts and when this data was being web scraped, the scraper had to be manually adjusted for each script. Even after these adjustments, there were inconsistencies regarding what counted as environmental text, what was metadata (ie. chapter titles, notes, comments), and what was actual character text. These inconsistencies occasionally pop up in the generated text output of the model and add

a degree of non-sensical randomness.

Chapter 6

Conclusion

NLG and NLU are both new fields that have great promise for the future. Despite the incredible performance of transformer-based architectures in NLU, there is still room to improve language models. This study furthermore highlights that better language models are the key to all types of understanding and generative tasks. Specifically, the application of a more sophisticated language model towards emotion classification could potentially produce not only a higher level of accuracy at classifying the ISEAR emotional data, but also at understanding contextually-based emotions that are currently unaccounted for such as sarcasm. Detecting emotions through an unsupervised clustering approach could prove to provide the most interesting results however as this would allow a model to ascribe its own categorization of emotion. This could even lead to better performance at tasks such as emotional style transferring, where results are entirely dependent on an accurate source profile.

One of the key areas this paper focuses on is not just text generation, but dynamic story generation. In the scope of this paper, this implies multiple levels of control in terms of guiding the story through environmental actions and deciding the characters that are present in a scene. Following these lines, it was hoped to make the environment interactive and for a user to actively guide conversation. While this was shown to be possible and a user could interject statements into the story that affected the generated responses of the other chatbots, text generation was incredibly intensive and took upwards of 15 or 30 seconds for each short response to be output running on a higher end NVIDA GTX 1080 GPU. Future work could involve evaluating approaches that are less computationally intensive and more suitable for generalized use such as in a web or mobile environment. However, with a proof of concept in hand, better performance is only a matter of effort rather than a fundamental rethinking.

The ability for chatbots to converse using natural language is a feat of modern language modeling techniques. However, the ability for chatbots to not only converse, but to also convey information has wide ranging applications. Future work using the techniques outlined in this paper are to shift towards more specific applications of emotional, question-answering models. One identified area of use is in the popular messaging app Discord. Discord has an easy to use API that allows bots to hook into a variety of different triggers. This makes it a natural environment for a chatbot to exist. Some of the possibilities on this platform include using it as a natural way to "search" chat history by taking advantage of its ability to answer contex-

tual questions in a channel, training it on domain specific datasets such as Stack Overflow so that it can answer programming related questions, and using it to emulate different character personas such as done in this paper. Additionally, Discord provides a hook that triggers every time a message is sent on a channel. Using this hook, specific commands can be searched for telling the bot to perform certain actions. The Discord bot that was created based on this paper includes commands for outputting real-time emotional breakdowns of a specified character or user on Discord using the ISEAR classification model, switching models and changing hyper-parameters for text generation, and is triggered based on keyword names being mentioned so that conversation can flow naturally. The bot also has the ability to trigger any of this commands and could therefore theoretically tune its own hyper-parameters and/or switch models, which poses an interesting ethical question about control.

The potential use cases for understanding and generating language are wide ranging. This paper explores some of the possibilities allowed by SOTA techniques to dramatically improve interactions in a controlled and directed manner over previous, non-generative and template-based approaches and shows how these techniques are the future of text generation. However, this only scratches the surface of what is possible and merely serves as one application along the path towards the future of deep learning-based language modeling.

Bibliography

[1] R. Stonebraker, *Dynamic Story Generation through Directed Chatbot Interactions*, version v0.1.0, Apr. 2020. DOI: 10.5281/zenodo.3762840. [Online]. Available: https://doi.org/10.5281/zenodo.3762840.

[2] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data", *arXiv:1705.02364 [cs]*, Jul. 2018, arXiv: 1705.02364. [Online]. Available: http://arxiv.org/abs/1705.02364.

[3] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations", *arXiv:1909.11942 [cs]*, Feb. 2020, arXiv: 1909.11942. [Online]. Available: http://arxiv.org/abs/1909.11942.

[4] *Emotion detection and recognition from text using deep learning*, Nov. 2015. [Online]. Available: https://devblogs.microsoft.com/cse/2015/11/29/emotion-detection-and-recognition-from-text-using-deep-learning/.

[5] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation", in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. [Online]. Available: https://www.aclweb.org/anthology/D14-1162.

[6] *Glove and fasttext — two popular word vector models in nlp - dzone ai*. [Online]. Available: https://dzone.com/articles/glove-and-fasttext-two-popular-word-vector-models.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding", *CoRR*, vol. abs/1810.04805, 2018. arXiv: 1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805.

[8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners", *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[9] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov, *Towards ai-complete question answering: A set of prerequisite toy tasks*, 2015. eprint: arXiv:1502.05698.

[10] J. S. Ball, "Using NLU in context for question answering: Improving on facebook's babi tasks", *CoRR*, vol. abs/1709.04558, 2017. arXiv: 1709.04558. [Online]. Available: http://arxiv.org/abs/1709.04558.

[11] J. Weston, "Dialog-based language learning", *CoRR*, vol. abs/1604.06045, 2016. arXiv: 1604.06045. [Online]. Available: http://arxiv.org/abs/1604.06045.

[12] J. Li, A. H. Miller, S. Chopra, M. Ranzato, and J. Weston, "Dialogue learning with human-in-the-loop", *CoRR*, vol. abs/1611.09823, 2016. arXiv: 1611.09823. [Online]. Available: http://arxiv.org/abs/1611.09823.

[13] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100, 000+ questions for machine comprehension of text", *CoRR*, vol. abs/1606.05250, 2016. arXiv: 1606.05250. [Online]. Available: http://arxiv.org/abs/1606.05250.

[14] D. Sarkar, *Implementing deep learning methods and feature engineering for text data: The continuous bag of words (cbow)*, Apr. 2018. [Online]. Available: https://www.kdnuggets.com/2018/04/implementing-deep-learning-methods-feature-engineering-text-data-cbow.html.

[15] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations", in *Proc. of NAACL*, 2018.

[16] J. Howard and S. Ruder, *Universal language model fine-tuning for text classification*, 2018. arXiv: 1801.06146 [cs.CL].

[17] M. S. Z. Rizvi, *Demystifying bert: The groundbreaking nlp framework*, Jan. 2020. [Online]. Available: https://www.analyticsvidhya.

com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need", *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762.

[19] K. R. Scherer and H. G. Wallbott, "Evidence for universality and cultural variation of differential emotion response patterning", *Journal of Personality and Social Psychology*, vol. 66, no. 2, pp. 310–328, 1994, ISSN: 1939-1315(Electronic),0022-3514(Print). DOI: 10.1037/0022-3514.66.2.310.